

**METHOD AND DEVICE FOR DEBUGGING PROCESS STEPPING TYPE PROGRAM**

Patent Number: JP2001154711  
Publication date: 2001-06-08  
Inventor(s): KUWA SHUNJI;; JINKAWA TAKESHI  
Applicant(s): OMRON CORP  
Requested Patent: ☐ JP2001154711  
Application Number: JP19990338692 19991129  
Priority Number(s):  
IPC Classification: G05B19/048; G05B19/05; G05B23/02  
EC Classification:  
Equivalents:

---

**Abstract**

---

**PROBLEM TO BE SOLVED:** To materialize a tracing function capable of verifying the execution state of a process stepping type program in a programmable logic controller at high visibility and detailed timing.

**SOLUTION:** The programmable controller distinguishes the activation/non-activation of the whole process stepping type program operating the activation/ non-activation in the execution of the program in the order of generating changes of activated state, both the activation and non-activation are stored in each operation, the stored information is read out, and the status changes are time-sequentially reflected to a patterned process stepping type program list by the display of an execution locus and a display format such as the continuous display of respective states.

---

Data supplied from the esp@cenet database - I2

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-154711

(P2001-154711A)

(43) 公開日 平成13年6月8日 (2001.6.8)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード(参考)
G 0 5 B	19/048	G 0 5 B 23/02	3 0 1 N 5 H 2 2 0
	19/05	19/05	D 5 H 2 2 3
	23/02		W
	3 0 1		

審査請求 未請求 請求項の数 9 O L (全 28 頁)

(21) 出願番号 特願平11-338692

(22) 出願日 平成11年11月29日 (1999.11.29)

(71) 出願人 000002945

オムロン株式会社

京都市下京区塩小路通堀川東入南不動堂町  
801番地

(72) 発明者 桑 俊司

京都府京都市右京区花園土堂町10番地 オ  
ムロン株式会社内

(72) 発明者 陣川 健

京都府京都市右京区花園土堂町10番地 オ  
ムロン株式会社内

(74) 代理人 100098899

弁理士 飯塚 信市

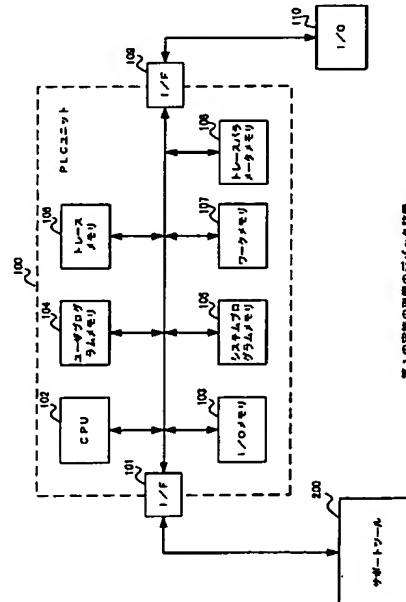
最終頁に続く

(54) 【発明の名称】 工程歩進型プログラムのデバッグ方法および装置

(57) 【要約】

【課題】 プログラマブルロジックコントローラにおける工程歩進型プログラムの実行状態を視認性が高く、かつ詳細なタイミングが検証できるようにしたトレース機能を実現する。

【解決手段】 プログラマブルロジックコントローラ側で工程歩進型プログラムの実行における活性化／非活性化を動作しているプログラム全体に対して、かつ活性化状態の変化が発生した順序通りに、活性化／非活性化を区別し、その両方を動作毎に記憶し、その記憶された情報を読み出し、その状態の変化を図示化された工程歩進型のプログラムリスト上に、実行の軌跡としての表示や、その時々々の状態の連続表示などの表示形態により、時系列に反映させる。



【特許請求の範囲】

【請求項1】 プログラマブルコントローラにおける工程歩進型プログラムの実行状態を監視する工程歩進型プログラムのデバッグ方法において、  
上記プログラマブルコントローラによる上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報を順次記憶する第1の工程と、

上記記憶した上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報を読み出す第2の工程と、

上記読出した上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報に基づいて、各工程での活性化状態となった工程を、図示化して表示された上記工程歩進型プログラム上にマッピングして表示する第3の工程とを有する、  
ことを特徴とする工程歩進型プログラムのデバッグ方法。

【請求項2】 第3の工程が、

上記読出した上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報に基づいて、各工程での活性化状態となった工程を、図示化して表示された上記工程歩進型プログラム上にマッピングして一括表示するものである、  
ことを特徴とする請求項1に記載の工程歩進型プログラムのデバッグ方法。

【請求項3】 第3の工程が、

上記読出した上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報に基づいて、各工程の活性化状態の変化を、図示化して表示された上記工程歩進型プログラム上にマッピングして変化順に表示するものである、  
ことを特徴とする請求項1に記載の工程歩進型プログラムのデバッグ方法。

【請求項4】 工程歩進型プログラムの実行を行うとともに、上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報を順次記憶する記憶手段を有するプログラマブルコントローラと、

上記プログラマブルコントローラに通信手段を介して接続され、上記プログラマブルコントローラによる工程歩進型プログラムの実行状態を表示して監視するデバッグ装置とを有し、

上記デバッグ装置には、  
上記記憶手段から上記通信手段を介して上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報を読み出す読出し手段と、

上記読出し手段により読み出された上記工程歩進型プログラムの実行時における各工程の活性化および／または

非活性化の状態変化を示す情報に基づいて、各工程の活性化状態の変化を、図示化して表示された上記工程歩進型プログラム上にマッピングして表示する表示制御手段と、

が設けられることを特徴とするプログラマブルコントローラシステム。

【請求項5】 工程歩進型プログラムの実行を行う機能を有するプログラマブルコントローラであって、

上記プログラマブルコントローラには、上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報を順次記憶する記憶手段が設けられていることを特徴とするプログラマブルコントローラ。

【請求項6】 工程歩進型プログラムの実行を行うとともに、上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報を順次記憶する記憶手段を有するプログラマブルコントローラに通信手段を介して接続され、上記プログラマブルコントローラによる工程歩進型プログラムの実行状態を表示して監視するデバッグ装置であって、

上記デバッグ装置には、  
上記記憶手段から上記通信手段を介して上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報を読み出す読出し手段と、

上記読出し手段により読み出された上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報に基づいて、各工程の活性化状態の変化を、図示化して表示された上記工程歩進型プログラム上にマッピングして表示する表示制御手段と、

が設けられることを特徴とするデバッグ装置。

【請求項7】 工程歩進型プログラムの実行を行う機能を有するプログラマブルコントローラであって、

上記プログラマブルコントローラには、  
上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報を順次記憶する記憶手段と、

上記記憶手段から上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報を読み出す読出し手段と、

上記読出し手段により読み出された上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報に基づいて、各工程の活性化状態の変化を、図示化して表示された上記工程歩進型プログラム上にマッピングして表示する表示制御手段と、

が設けられることを特徴とするプログラマブルコントローラ。

【請求項8】 表示制御手段が、各工程での活性化状態と

なった工程を、図示化して表示された上記工程歩進型プログラム上にマッピングして一括表示することを特徴とする請求項4、6、及び7のいずれかに記載のシステム又は装置。

【請求項9】 表示制御手段が、各工程の活性状態の変化を、図示化して表示された上記工程歩進型プログラム上にマッピングして変化順に表示することを特徴とする請求項4、6、及び7のいずれかに記載のシステム又は装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、プログラマブルコントローラにおける工程歩進型プログラムの実行状態を表示して監視する工程歩進型プログラムのデバッグ方法および装置に関する。

【0002】

【従来の技術】一般に、プログラマブルコントローラのプログラミング言語としては、ラダーダイアグラムが主流であるが、最近、SFC(Sequential Function Chart)のような工程歩進型のプログラム言語も普及し始めている。

【0003】この工程歩進型のプログラムを実行可能とする処理方式を採用するプログラマブルコントローラとしては、特許番号2621631号に開示されたものが知られている。

【0004】また、プログラマブルコントローラにおいて、ユーザが記述したプログラムが正しく動作するかを確認するための手法としては、ラダーダイアグラムの場合、リレーと呼ばれるビット情報やチャネルデータのモニタリングや、その変化を時系列的に記憶し、その後でタイムチャート形式で表示する手法などが知られている。

【0005】これはラダーダイアグラムが、入力状態とリレーやチャネルの状態の組み合わせを状態変化の条件として、その変化をリレーやチャネルの状態を変更していくことでその制御アルゴリズムが形成されているため、逆にラダーダイアグラムがリレーやデータの状態で制御状況の変化を把握することができるためである。

【0006】これに対して、工程歩進型プログラムは、各工程での動作と次工程への移行条件を組み合わせる動作の流れを表現することを基本とし、条件分岐や複数の工程の同時実行などを繋げて制御アルゴリズムを形成していくため、プログラムの動作確認では、どの工程が動作しているのか、どのような順序でいつ状態が変化したのかを把握することが必要となってくる。

【0007】ところで、従来、工程歩進型の言語で記述されたプログラムの動作確認(デバッグ)の手法としては、以下に示す手法が一般的である。

【0008】1)監視したい対象を指定して工程の動作状態(活性状態)をモニタする。

【0009】2)図示化されたプログラムリスト上で現在活性化している工程を反転表示させ更新していく。

【0010】ただし、これらの手法は、工程間の遷移速度が比較的緩やかで、人間が目視してその変化を認識できるような場合にのみ有効とある。そこで、工程間の遷移が頻繁に起こる場合は、活性状態の変化をその発生の都度コントローラ側で記憶し、後にそれをデバッグ装置に読み出してオペレータが解析する工程歩進動作のトレース手法が有効になる。

10 【0011】

【発明が解決しようとする課題】この発明が解決しようとする課題は、上述したような工程間の遷移が頻繁に起こる工程歩進型プログラムの動作確認の手法としての工程歩進動作のトレースについてである。

【0012】すなわち、従来のこの種の手法においては、工程歩進型プログラムの活性状態の変化を工程番号によりその発生順に記憶し、かつ複数のプログラムブロック間の同期を実現しているが、この場合、オペレータがそのトレース結果を解析するのに工程番号の羅列を用いるしかなく、その認識性の低さから解析作業に多大な困難をきたすことが多かった。

20

【0013】また、複数のチャート間のタイミング検証が必要な際にも、トレース結果の同期はとれているものの、個別の工程毎の状態変化までは正確に認識することは難しかった。

【0014】また、動作の重要な要素である工程が非活性化したタイミングが明示的に記憶されていないため、外部からの活性状態の変化が混在するような場合は、詳細な状態把握が困難になった。

30

【0015】そこで、この発明は、プログラマブルコントローラにおける工程歩進型プログラムの実行状態を視認性が高く、かつ詳細なタイミングが検証できるようにしたトレース機能を実現することを目的とする。

【0016】

【課題を解決するための手段】この発明の工程歩進型プログラムのデバッグ方法および装置は、SFC等の工程歩進型プログラムの実行において活性、または非活性に変化したステップをサンプリングするトレース機能を実現する。

40

【0017】すなわち、プログラマブルコントローラ側で工程歩進型プログラムの実行における活性化/非活性化を動作しているプログラム全体に対して、かつ活性状態の変化が発生した順序通りに活性化/非活性化を区別してその両方を各動作毎にサンプリングして記憶し、その記憶された情報を読み出し、その状態の変化を図示化された工程歩進型のプログラムリスト上に実行の軌跡としての表示や、その時々状態の連続表示などの表示形態により、時系列に反映させる。

50

【0018】これにより、プログラマブルコントローラにおける工程歩進型プログラムの実行状態を視認性が高

く、かつ詳細なタイミングが検証できるようにしたトレース機能を実現する。

【0019】ここで、上記サンプリングするタイミングはステップ活性状態の変化時（活性遷移、命令／外部からの強制変更）で、ステップが活性から非活性または非活性から活性に状態遷移するタイミングでトレースバッファサンプリングを行う。

【0020】このサンプリングした結果表示は、パターンAとパターンBの2通りの形式で行うことができる。

【0021】パターンAでは、活性状態になったステップを全て反転表示する。つまり、チャート上の実行軌跡を表示することで選択分岐箇所をどちらの経路で通過したかを一目で把握することができる。

【0022】パターンBでは、ユーザが指定するタイミング（キーアクション若しくは一定間隔）で、表示画面を順次切り替えながら表示する。つまり、活性状態の変化から次の変化までの時間を圧縮してその変化をトレースすることができる。

【0023】すなわち、この発明の工程歩進型プログラムのデバッグ方法は、プログラマブルコントローラにおける工程歩進型プログラムの実行状態を監視する工程歩進型プログラムのデバッグ方法において、上記プログラマブルコントローラによる上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報を順次記憶する第1の工程と、上記記憶した上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報を読み出す第2の工程と、上記読出した上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報に基づいて、各工程の活性状態の変化を、図示化して表示された上記工程歩進型プログラム上にマッピングして表示する第3の工程とを有する。

【0024】ここで、好ましい実施の形態では、第3の工程としては、上記読出した上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報に基づいて、各工程での活性状態となった工程を、図示化して表示された上記工程歩進型プログラム上にマッピングして一括表示する（パターンA）ように構成することができる。

【0025】また、好ましい他の実施の形態では、第3の工程としては、上記読出した上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報に基づいて、各工程の活性状態の変化を、図示化して表示された上記工程歩進型プログラム上にマッピングして変化順に表示する（パターンB）ように構成することができる。

【0026】ここで、上記機能は、プログラマブルコントローラとこのプログラマブルコントローラに接続されたサポートツール（デバッグ装置として機能する）とか

らなるプログラマブルコントローラシステムにより実現することができる。

【0027】すなわち、この発明のプログラマブルコントローラシステムは、工程歩進型プログラムの実行を行うとともに、上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報を順次記憶する記憶手段を有するプログラマブルコントローラと、上記プログラマブルコントローラに通信手段を介して接続され、上記プログラマブルコントローラによる工程歩進型プログラムの実行状態を表示して監視するデバッグ装置とを有する。

【0028】上記デバッグ装置には、上記記憶手段から上記通信手段を介して上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報を読み出す読出し手段と、上記読出し手段により読み出された上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報に基づいて、各工程の活性状態の変化を、図示化して表示された上記工程歩進型プログラム上にマッピングして表示する表示制御手段と、が設けられる。

【0029】また、上記機能は、プログラマブルコントローラ単独でも実現することができる。

【0030】すなわち、この発明のプログラマブルコントローラは、工程歩進型プログラムの実行を行う機能を有するプログラマブルコントローラであって、上記プログラマブルコントローラには、上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報を順次記憶する記憶手段と、上記記憶手段から上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報を読み出す読出し手段と、上記読出し手段により読み出された上記工程歩進型プログラムの実行時における各工程の活性化および／または非活性化の状態変化を示す情報に基づいて、各工程の活性状態の変化を、図示化して表示された上記工程歩進型プログラム上にマッピングして表示する表示制御手段と、が設けられる。

【0031】

【発明の実施の形態】以下、この発明に係わる工程歩進型プログラムのデバッグ方法および装置の実施の形態を添付図面を参照して詳細に説明する。

【0032】この発明に係わる工程歩進型プログラムのデバッグ方法の実現手法としては、プログラマブルコントローラ（以下、PLCという）とこのPLCに接続されたサポートツールとによる実現手法とPLC単独で行う実現手法との2つの手法が考えられる。

【0033】そこで、以下の説明においては、前者の実現手法を第1の実施の形態とし、後者の実現手法を第2の実施の形態として説明する。

【0034】図1は、この発明に係る工程歩進型プログラムのデバッグ方法の第1の実施の形態を示すブロック図である。

【0035】図1に示す第1の実施の形態の工程歩進型プログラムのデバッグ方法は、PLCのCPUユニット100にデバッグ装置として機能するサポートツール200を接続したプログラマブルコントローラシステムとして実現される。

【0036】ここで、CPUユニット100は、サポートツール200とのインタフェースをなすシリアルインタフェース(I/F)101、演算プロセッサ(CPU)102、入出力メモリ(I/Oメモリ)103、ユーザプログラムメモリ104、システムプログラムメモリ105、トレースメモリ106、ワークメモリ107、トレースパラメータメモリ108、入出力装置(I/O)110とのインタフェースをなす入出力インタフェース(I/F)109を具備して構成され、シリアルインタフェース101にサポートツール200が接続され、入出力インタフェース109に入出力装置110が接続される。

【0037】ここで、CPU102は、このPLCユニット100の全体動作を統括制御するものである。

【0038】また、システムプログラムメモリ105には、CPU102で処理されるシステムプログラムが記憶され、ユーザプログラムメモリ104には、ユーザプログラムが記憶される。

【0039】また、トレースメモリ106には、このトレース機能を実現するためのサンプリング結果であるトレースデータが記憶される。

【0040】また、トレースパラメータメモリ108には、このトレース機能を実行するための各種トレースパラメータが記憶される。

【0041】このトレースパラメータメモリ108に記憶されたトレースパラメータとしては、

- 1) トレース対象チャート番号
  - 2) 接点およびステップ番号などのトレースのトリガ条件
  - 3) トレースのサンプリングディレイ値
- などが挙げられる。

【0042】ワークメモリ107には、CPU102の処理全般にわたる各種データが記憶される。

【0043】また、I/Oメモリ103は、CPU102によるプログラム実行により入出力装置110に対して入出力される入出力情報の代用として参照/書き換えられるメモリである。

【0044】また、サポートツール200は、

- 1) トレース処理のための各種トレースパラメータの生成
- 2) CPU102に対するコマンド発行(トレース実行依頼)

3) トレース結果の解析と表示等を行う。

【0045】図2は、図1に示したサポートツール200の詳細構成を示すブロック図である。

【0046】図2において、このサポートツール200は、表示装置260の表示を制御するための表示制御用メモリ210、このトレース解析表示を制御するための各種情報を記憶するメモリ空間を構成するメモリ220、演算装置(CPU)230、バス240、インタフェース(I/F)250を具備して構成され、表示制御用メモリ210には、表示装置260が接続され、バス240には入力装置270、表示制御用メモリ210、メモリ220、演算装置230、インタフェース250が接続され、インタフェース250には、CPUユニット100が接続される。

【0047】ここで、中央演算装置230は、このサポートツール200の全体動作を統括制御するものである。

【0048】また、メモリ220には、

- 1) トレースデータ用データエリア221
- 2) ユーザプログラムデータエリア222
- 3) サポートツールプログラムモジュール223

が設けられる。

【0049】ここで、トレースデータ用データエリア221には、入出力装置200からのトレースデータ読み出しコマンドの入力により、インタフェース(I/F)250を介してCPUユニット100のトレースメモリ106から読み出されるトレースデータが格納される。

【0050】また、ユーザプログラムモジュール222には、CPUユニット100のユーザプログラムメモリ104に記憶されるSFCで記述されたユーザプログラムと等価なプログラムデータが記憶される。

【0051】また、サポートツールプログラムモジュール223には、演算装置230が実行するこのトレース機能を実行するためのプログラムが格納される。

【0052】また、バス240は、演算装置230が、メモリ220、入力装置270、表示制御用メモリ210をアクセスするためのインタフェースを構成する。

【0053】また、インタフェース250はCPUユニット100と通信を行うためのインタフェースである。

【0054】また、入力装置270は、ユーザの要求を得るための装置で、キーボード、ポインティングデバイス等から構成される。

【0055】また、表示装置260は、表示制御用メモリ210の状態を表示する装置で、CRT等から構成される。

【0056】さて、この発明の工程歩進型プログラムのデバッグ装置は、SFC実行において活性、または非活性に変化したステップをサンプリングするトレース機能を実現する。

【0057】ここで、サンプリングするタイミングは、ステップ活性状態の変化時、すなわち、ステップの活性遷移、命令／外部からの強制変更時で、ステップが活性から非活性または非活性から活性に状態遷移するタイミングでトレースバッファサンプリングを行う。

【0058】このサンプリングした結果は、サポートツール200において後に詳述する2種類の形式で表示される。

【0059】上記第1の形態のデバッグ装置の動作の概略を示すと以下になる。

【0060】1) サポートツール200からCPUユニット100に、SFCトレースのためのトレース対象のSFCチャート、トリガ条件、ディレイ値等のトレースパラメータが設定される。

【0061】2) その後、サポートツール200からCPUユニット100にトレース開始の指示が発行される。これを機会にCPUユニット100はSFC実行における活性状態の変化をトレースメモリ(トレースバッファ)106に記憶し始める(データサンプリング)。

【0062】3) トリガ条件が成立した時点から、上記ディレイ値をふまえてトレースバッファが記憶データで埋まるまでサンプリングは続けられる。

【0063】4) サンプリングが完了すると、CPUユニット100はサポートツール200にトレース完了を知らせる。

【0064】5) サポートツール200は、CPUユニット200のトレースバッファの内容を読み出し、ユーザが指定する結果表示形式にてサンプリング結果を表示する。なお、サンプリング完了前のアボートにおいてもその時点までの結果を表示可能である。

【0065】すなわち、予めサポートツール200にて、所定のトレースパラメータが設定され、SFCトレースが起動されると、後述するように、SFCチャート上で活性状態の変化が起こり、それと並行してSFCトレースが実行され、その結果をサポートツール200が読み出し、その結果を表示する。

【0066】この結果表示には、パターンAとパターンBの2通りの形式で行うことができる。なお、活性状態の変化は、ステップ間の遷移条件の成立に依存するため、タイミングは不定である。

【0067】ここで、パターンAでは、例えば、活性状態になったステップを全て反転表示する。つまり、チャート上の実行軌跡を表示することで選択分岐箇所をどちらの経路で通過したかを一目で把握することができる。

【0068】また、パターンBでは、例えば、ユーザが指定するタイミング(キーアクション若しくは一定間隔)で、活性状態になったステップを順次切り替えながら表示する。つまり、活性状態の変化から次の変化までの時間を圧縮してその変化をトレースすることができる。

【0069】この表示パターンにおいては、以下の操作上の考慮を行い、ユーザの視認性、操作性を向上させることができる。

【0070】すなわち、このパターン表示において、以下の表示機能を付加することにより、より視認性の高いユーザインタフェースを提供することもできる。

【0071】1) ステップの変化点を点滅させ強調する。

【0072】2) チャートが表示画面より大きく、1画面で表示しきれず、活性状態の遷移により該当箇所が画面より外れてしまうような場合には、活性箇所の移動に応じて画面を自動的に切り替える。

【0073】図3は、この発明に係わる工程歩進型プログラムのデバッグ装置における工程歩進型プログラムの表示例を示す図である。

【0074】図3において、この表示例は、サポートツール200の表示装置260の表示画面上に表示されるSFCプログラムの一例を示している。

【0075】すなわち、図3においては、SFCプログラムのステップS18からステップS29までが図示化して示されており、このSFCプログラムには、ステップS18の後に、ステップS19、ステップS20、ステップS21への第1の分岐を有しており、また、ステップS22の後に、ステップS24、S25の第2の分岐を有しており、ステップS27、ステップS23、ステップS28は、ステップS29の前で統合されている。

【0076】さて、この第1の実施の形態の工程歩進型プログラムのデバッグ装置におけるパターンAの表示例においては、サポートツール200の表示装置260の表示画面上に表示される図3に示したSFCプログラムチャート上で、活性状態になったステップを一括してマッピングして表示することで該SFCプログラムのトレース結果を表示する。

【0077】図4は、この発明に係わる工程歩進型プログラムのデバッグ装置におけるパターンAの表示例を示す図である。

【0078】すなわち、図4においては、図3にしたSFCプログラムチャート上で、活性状態になったステップを全て反転表示することで、該SFCプログラムのトレース結果を表示している。

【0079】すなわち、図4において、ハッチングで示すように、ステップS18～ステップS23、およびステップS25～ステップS29が、このSFCプログラムのトレースにより活性化したステップを示している。

【0080】図4から明らかなように、このSFCプログラムチャート上で、ステップS22の後の第2の分岐点においては、ステップS25が選択されたことが視覚的に容易に認識できる。

【0081】つまり、このパターンAの結果表示によれ

ば、SFCプログラムチャート上の実行軌跡から選択分岐箇所をどちらの経路で通過したかを一目で把握することができる。

【0082】また、この第1の実施の形態の工程歩進型プログラムのデバッグ装置におけるパターンBでは、サポートツール200の表示装置260の表示画面上に表示される図3に示したSFCプログラムチャート上で、

10 活性化状態になったステップをユーザが指定するタイミングで、順次マッピングしてサポートツール200の表示装置260の表示画面を順次切り替えながら表示する。

【0083】ここで、上記表示画面を切り替えるタイミングは、ユーザによるサポートツール200の入力装置270におけるキアクション若しくは一定間隔で行われる。

【0084】つまり、SFCプログラムの各ステップの活性化状態の変化から次の変化までの時間を圧縮してその変化をトレースして表示する。

【0085】図5乃至図13は、この発明に係わる工程歩進型プログラムのデバッグ装置におけるパターンBの表示例を示す図である。

【0086】すなわち、図5は、ステップS18が活性化されている第1の表示画面を示しており、図6は、次に、ステップS19、ステップS20、ステップS21が活性化された第2の表示画面を示しており、図7は、次に、ステップS22、ステップS20、ステップS21が活性化された第3の表示画面を示しており、図8は、次に、ステップS22、ステップS23、ステップS21が活性化された第4の表示画面を示しており、図9は、次に、ステップS22、ステップS23、ステップS26が活性化された第5の表示画面を示しており、図10は、次に、ステップS22、ステップS23、ステップS28が活性化された第6の表示画面を示しており、図11は、次に、ステップS25、ステップS23、ステップS28が活性化された第7の表示画面を示しており、図12は、次に、ステップS27、ステップS23、ステップS28が活性化された第8の表示画面を示しており、図13は、次に、ステップS29が活性化された第9の表示画面を示している。

【0087】このパターンBのトレース結果表示によれば、図5乃至図13から明らかなように、SFCプログラムチャート上で、各ステップの活性化状態の変化が発生した順をオペレータの望む時間感覚で時系列的に再現でき、また、活性化状態の遷移の狭間に、強制的な活性化状態の変更が混入したとしてもそれを区別してかつ時系列的に確認できる。

【0088】次に、図1および図2に示したこの第1の実施の形態のデバッグ装置の概略動作を説明する。

【0089】図14は、図1および図2に示したデバッグ装置の全体動作を説明するための全体フローを示すフローチャートである。

【0090】図14において、左側のフローがサポートツール200側の処理を示し、右側のフローがPLCのCPUユニット100側の処理を示す。

【0091】サポートツール200において、入力装置270からオペレータよりトレース実行に必要なパラメータが入力されると（ステップ1401）、このトレース実行に必要なパラメータはPLCのCPUユニット100に送信される。

10 【0092】CPUユニット100は、このサポートツール200から送信されたトレース実行に必要なパラメータを受信し、この受信したトレース実行に必要なパラメータをトレースパラメータメモリ108に記憶する（ステップ1411）。

【0093】また、オペレータは、上記トレース実行に必要なパラメータを入力装置270から入力した後、サポートツール200の入力装置270からトレース開始入力を行う（ステップ1402）。これにより、サポートツール200からトレース開始指示がCPUユニット100に送信される。

20 【0094】CPUユニット100では、サポートツール200からのトレース開始指示があるかを調べており（ステップ1402）、ここで、サポートツール200からのトレース開始指示がない場合は（ステップ1412NO）、ステップ1412に戻り、サポートツール200からのトレース開始指示を待つが、ステップ1412でサポートツール200からのトレース開始指示があると判断されると（ステップ1412YES）、トレース処理を開始する（ステップ1414）。

30 【0095】このトレース処理は、SFC実行処理中に行われ、活性化状態の遷移が発生したとき、活性化したステップ（工程）の番号並びに非活性化したステップの番号をトレースメモリ106に順次記憶していくことにより行われる（ステップ1414）。

【0096】また、別途、トリガ条件、例えば、任意のリレーの立ち上がり、任意のステップの活性等の成立を検知し、このトリガ条件が成立したタイミングから予め設定されたサンプリングディレイ値を取ったトレースメモリ106が記憶データで埋まるまでサンプリング、すなわち、トレースメモリ106へのデータの記憶を続ける。

40 【0097】このトレースの完了は、ステップ1415で判断され、トレースが完了していない場合は（ステップ1415NO）、ステップ1414に戻り、上記トレースを続けるが、ステップ1415でトレース完了と判断されると（ステップ1416YES）、トレース完了通知を行う（ステップ1416）。

50 【0098】このステップ1416のトレース完了通知は、サポートツール200によるトレース完了確認（ステップ1403）に基づく完了問合せのレスポンスとして行われる。



【0099】すなわち、サポートツール200は、ステップ1403でトレース完了の確認を行い、トレースが完了していないと判断されると(ステップ1404NO)、ステップ1403に戻り、再度トレース完了の確認を行うが、ステップ1404でトレース完了と判断されると(ステップ1404YES)、CPUユニット100のトレースメモリ106の読出し処理を実行し(ステップ1405)、CPUユニット100に対してトレースメモリ送信依頼を通知する。

【0100】CPUユニット100では、トレース完了通知(ステップ1416)を行った後、サポートツール200からのトレースメモリ送信依頼があるかを調べており(ステップ1417)、ここで、サポートツール200からのトレースメモリ送信依頼がないと判断されると(ステップ1417NO)、ステップ1417に戻り、サポートツール200からのトレースメモリ送信依頼を待つが、ステップ1417で、サポートツール200からのトレースメモリ送信依頼があると判断されると(ステップ1417YES)、トレースメモリ106に記憶したデータをサポートツール200に送信し(ステップ1418)、このトレース処理を完了する。

【0101】また、サポートツール200では、トレースメモリ読出し処理(ステップ1405)により、CPUユニット100から送信されたトレースメモリ106に記憶したデータを解析し(ステップ1406)、この解析結果に基づき前述したパターンA若しくはパターンBでトレース結果表示を行い(ステップ1407)、このトレース処理を完了する。

【0102】次に、CPUユニット100およびサポートツール200における詳細処理について説明する。

【0103】図15は、図1に示したCPUユニット100の全体処理フローを示すフローチャートである。

【0104】図15において、PLCの電源が投入(ON)されると、まず、所定の共通処理を実行し(ステップ1501)、次に、このPLCがプログラム運転可能かを調べる(ステップ1502)。ここで、プログラム運転可能と判断されると(ステップ1502YES)、このPLCのプログラム運転処理を実行する(ステップ1503)。

【0105】また、ステップ1502でプログラム運転不可と判断された場合(ステップ1502NO)、若しくはステップ1503のプログラム運転処理が終了すると、次に、I/Oメモリ103をリフレッシュするI/Oリフレッシュ処理を実行し(ステップ1504)、その後、サポートツール200との通信を行い(ステップ1505)、ステップ1501に戻り、上記処理を繰り返す。

【0106】ここで、この第1の実施の形態におけるCPUユニット100のトレースメモリ106の仕様について説明する。

【0107】図16は、図1に示したCPUユニット100のトレースメモリ106の仕様を示す図である。

【0108】図16において、トレースメモリ106には、ビットA、ビットB、SFCステップ番号(SFC STEP番号)を1レコードとしてサンプリングデータが記憶される。

【0109】ここで、ビットAは、1回の活性状態変化の終了フラグを示し、ビットBは、活性/非活性になったステップ種別を示す。なお、ビットBにおいて“1”は活性、“0”は非活性を示す。

【0110】また、1レコードは、活性状態が変化した1ステップ分(1W)とする。また、活性状態の変化、すなわち遷移等は、1度に複数ステップ分まとめて行われるので、この単位はビットAのフラグによって判断され、この単位が1回のサンプリングデータとなる。

【0111】図17は、図1に示したCPUユニット100におけるトレースサンプリング処理フローを示すフローチャートである。

【0112】図17において、まず、該当ステップが活性化しているかを調べる(ステップ1701)。ここで、該当ステップが活性化していないと判断されると(ステップ1701NO)、該当ステップの非活性化処理を実行する(ステップ1702)。

【0113】そして、次に、該当ステップはトレース中かを調べ(ステップ1703)、トレース中であると判断されると(ステップ1703YES)、トレースポイントが示すトレースメモリ106のアドレスに、ビットBをOFF(“0”)にして該当ステップのステップ番号を格納し(ステップ1704)、トレースポイントを次に更新する(ステップ1708)。

【0114】また、ステップ1701において、該当ステップが活性化していると判断されると(ステップ1701YES)、該当ステップの活性化処理を実行する(ステップ1705)。

【0115】そして、次に、該当ステップはトレース中かを調べ(ステップ1706)、トレース中であると判断されると(ステップ1706YES)、トレースポイントが示すトレースメモリ106のアドレスに、ビットBをON(“1”)にして該当ステップのステップ番号を格納し(ステップ1707)、トレースポイントを次に更新する(ステップ1708)。

【0116】ステップ1708でトレースポイントを次に更新した後、ステップ1703でトレース中ではないと判断された場合(ステップ1703NO)、ステップ1706でトレース中ではないと判断された場合(ステップ1706NO)は、次に、残ステップはあるかを調べる(ステップ1709)。

【0117】ここで、残ステップがあると判断されると(ステップ1709YES)、ステップ1701に戻り上記処理が繰り返される。

【0118】ステップ1709で、残ステップがないと判断されると(ステップ1709NO)、次に、該当ステップはトレース中かを調べ(ステップ1710)、トレース中である場合は(ステップ1710YES)、トレースポインタが示すアドレスの1レコード前のレコードのビットAをONにして(ステップ1711)、このトレースサンプリング処理を終了する。

【0119】また、ステップ1710で、該当ステップはトレース中でないと判断されると(ステップ1710NO)、このままこのトレースサンプリング処理を終了する。

【0120】図18は、図2に示したサポートツール200の全体処理フローを示すフローチャートである。

【0121】図2に示したサポートツール200は、

- 1) プログラム編集機能
  - 2) この発明に係わるトレース機能
  - 3) モニタ機能
- 等の多種の機能を有している。

【0122】そこで、図18に示したサポートツール200の全体処理フローにおいては、オペレータの選択操作により各機能への分岐処理を実行する。

【0123】すなわち、図18において、まず、所定の起動時初期処理を実行し(ステップ1801)、次に、オペレータによる機能選択を行う(ステップ1802)。

【0124】そして、次に、ステップ1802でオペレータにより選択された機能はプログラム編集機能かを調べ(ステップ1803)、プログラム編集機能が選択されていると判断された場合は(ステップ1803YES)、プログラム編集機能を実行する(ステップ1804)。

【0125】ステップ1803で、プログラム編集機能が選択されていないと判断された場合は(ステップ1803NO)、次に、トレース機能が選択されたかを調べる(ステップ1805)。

【0126】ここで、トレース機能が選択されていると判断された場合は(ステップ1805YES)、トレース機能を実行する(ステップ1807)。

【0127】また、ステップ1805でトレース機能が選択されていないと判断された場合は(ステップ1805NO)、次に、モニタ機能が選択されているかを調べる(ステップ1808)。ここで、モニタ機能が選択されていると判断された場合は(ステップ1808YES)、モニタ機能を実行し(ステップ1809)、モニタ機能が選択されていないと判断された場合は(ステップ1808NO)、以下同様にしてこのサポートツール200の各機能への分岐を行う(ステップ1810)。

【0128】次に、上記機能分岐処理が終了したかを調べ(ステップ1811)、終了していないと判断された場合は(ステップ1811NO)、ステップ1802へ

戻り、上記処理が繰り返されるが、ステップ1811で、この機能分岐処理が終了したと判断されると(ステップ1811YES)、所定の終了処理を実行し(ステップ1812)、このサポートツール200の全体処理を終了する。

【0129】図19は、図2に示したサポートツール200のトレース機能全体フローを示すフローチャートである。

【0130】ここで、この第1の実施の形態のデバッグ装置において用意されているサポートツール200とCPUユニット100との間のメッセージサービスコマンドについて説明する、この第1の実施の形態のデバッグ装置においては、サポートツール200とCPUユニット100との間で以下に示すメッセージサービスコマンドが用意されている。

【0131】1) トレースパラメータ設定

このトレースパラメータ設定コマンドは、送信コマンドのパラメータとして、トレース対象チャート番号、接点およびステップ番号などのトリガ、サンプリングディレイ値を含み、受信レスポンスのパラメータとして、正常終了またはエラー終了を含む。

【0132】2) トレース実行

このトレース実行コマンドは、送信コマンドのパラメータはないが、受信レスポンスのパラメータとして、正常終了またはエラー終了を含む。

【0133】3) トレースアボート(中止)

このトレースアボートコマンドは、送信コマンドのパラメータはないが、受信レスポンスのパラメータとして、正常終了またはエラー終了を含む。

【0134】4) トレース実行状態問い合わせ

このトレース実行状態問い合わせコマンドは、送信コマンドのパラメータはないが、受信レスポンスのパラメータとして、実行中または終了を含む。

【0135】5) トレースデータ読み出し

このトレースデータ読み出しコマンドは、送信コマンドのパラメータとして、トレースメモリアドレス、読み出しデータ長を含み、受信レスポンスのパラメータとして、トレースデータを含む。

【0136】さて、図19に示すフローチャートの処理を、図2を参照して説明すると、まず、オペレータによる入力装置270からのトレースパラメータの入力が行われる(ステップ1901)。

【0137】ここでトレースパラメータとは、上述したように

- 1) トレースの対象
- 2) トレースの開始条件(トリガ、サンプリングディレイ値)

等である。このトレースパラメータは、演算装置230に入力される。

【0138】次に、演算装置230からインタフェース

250を介してメッセージサービスコマンドを用いてこのトレースパラメータを設定したトレースパラメータ設定コマンドをCPUユニット100へ送信する(ステップ1902)。

【0139】次に、オペレータにより入力装置270からトレース実行を選択し、このトレース実行の指示を演算装置230へ入力し、演算装置230はトレース実行処理を実行する(ステップ1903)。

【0140】これにより、演算装置230からインタフェース250を介してメッセージサービスコマンドを用いてトレース実行コマンドをCPUユニット100へ送信する(ステップ1903)。

【0141】次に、トレース実行コマンドのレスポンスからCPUユニット100においてトレース中断であるか否かを判断する(ステップ1905)。

【0142】ここで、トレース中断中であると判断されると(ステップ1905YES)、演算装置230からインタフェース250を介してメッセージサービスコマンドを用いてトレースアポートコマンドをCPUユニット100へ送信する(ステップ1907)。

【0143】また、ステップ1905で、トレース中断中でないと判断されると(ステップ1905NO)、演算装置230からインタフェース250を介してメッセージサービスコマンドを用いてトレース実行状態問い合わせコマンドをCPUユニット100へ送信する(ステップ1906)。

【0144】そして、次に、このトレース実行状態問い合わせコマンドのレスポンスからトレース実行完了かを調べ(ステップ1908)、トレース実行完了でないと判断されると(ステップ1908NO)、ステップ1905に戻るが、トレース実行完了と判断されると(ステップ1908YES)、ステップ1909に進む。

【0145】ステップ1909では、演算装置230からインタフェース250を介してメッセージサービスコマンドを用いてトレースデータ読み出しコマンドをCPUユニット100へ送信する。そして、演算装置230は、このトレースデータ読み出しコマンドのレスポンスとしてCPUユニット100からトレースデータを受信し、この受信したトレースデータをメモリ220のトレースデータ用データエリア221に格納する。

【0146】次に、演算装置230は、メモリ220のユーザプログラムデータエリア222に格納されたユーザプログラムと等価なプログラムデータと、CPUユニット100のユーザプログラムとの一致をインタフェース250を介してメッセージサービスコマンドを用いて確認する(ステップ1911)。

【0147】ここで、プログラムの一致が確認されると(ステップ1911YES)、CPUユニット100のユーザプログラムメモリ104からユーザプログラムを読み出し(ステップ1912)、メモリ220のユーザ

プログラムデータエリア222のデータを元に表示制御用メモリ210にSFCチャートに相当するデータを書き込むSFCチャートの表示処理を実行する(ステップ1913)。

【0148】次に、オペレータによる入力装置270からのトレース表示範囲が選択されると、その選択結果を演算装置230に入力する(ステップ1914)。

【0149】続いて、オペレータによる入力装置270からトレース結果表示のパターンがパターンAかパターンBかが選択されると、その選択結果を演算装置230に入力する(ステップ1915)。

【0150】次に、演算装置230は、ステップ1915で、パターンAとパターンBとのどちらのパターンが選択されたかを調べ(ステップ1916)、ここで、パターンAが選択されていると判断されると(ステップ1916A)、メモリ220のトレースデータ用データエリア221に格納されたトレースデータを用いてパターンAの解析表示処理を実行し(ステップ1917)、パターンBが選択されていると判断されると(ステップ1916B)、メモリ220のトレースデータ用データエリア221に格納されたトレースデータを用いてパターンBの解析表示処理を実行し(ステップ1918)、このトレース機能全体フローを終了する。

【0151】図20は、図19示したパターンAの解析実行処理の詳細を示すフローチャートである。

【0152】このパターンAの解析実行処理は、サポートツール200の表示装置260の表示画面上でSFCプログラムチャートの活性状態になったステップ全てを一括して反転表示するものである。

【0153】図20において、サポートツール200の演算装置230は、まず、対象ステップ抽出処理を実行する(ステップ2001)。この対象ステップ抽出処理は、図19のステップ1914で指定された範囲に含まれるステップ番号をメモリ220のユーザプログラムデータエリア222のデータから抽出することにより行われる。

【0154】次に、演算装置230は、トレースデータから対象ステップのレコードを抽出する処理を実行する(ステップ2002)。このトレースデータから対象ステップのレコードを抽出する処理は、図19のステップ1914で指定された範囲に含まれるステップ番号を含まないトレースデータのレコードをメモリ220のトレースデータ用データエリア221から削除することにより行われる。

【0155】続いて、演算装置230は、フラグBが“1”のレコードを抽出する処理を実行する(ステップ2003)。このフラグBが“1”のレコードを抽出する処理は、フラグBが“0”(ステップ種別が非活性)のレコードをメモリ220のトレースデータ用データエリア221から削除することにより行われる。

【0156】次に、演算装置230は、抽出されたデータのステップ番号を順に読み出し、活性状態の表示を行う表示処理を実行する（ステップ2004）。この表示処理は、表示制御用メモリ210において、トレースデータ用データエリア221に含まれるステップ番号のステップに該当するデータを活性状態を意味するデータに、トレースデータ用データエリア221のレコード数書き換えることにより行われる。

【0157】そして、オペレータにより入力装置270の、例えば、SPACEキーが押下されると、表示装置260の画面変化が停止し（ステップ2005）、このパターンAの解析実行処理が終了する。

【0158】図21は、図19示したパターンBの解析実行処理の詳細を示すフローチャートである。

【0159】このパターンBの解析実行処理は、サポートツール200の表示装置260の表示画面上でSFCプログラムチャートの活性状態になったステップを順次反転表示しながら表示画面を切り替えて表示するものである。

【0160】図21において、サポートツール200の演算装置230は、まず、対象ステップ抽出処理を実行する（ステップ2101）。この対象ステップ抽出処理は、図19のステップ1914で指定された範囲に含まれるステップ番号をユーザプログラムデータエリア222のデータから抽出することにより行われる。

【0161】次に、演算装置230は、トレースデータから対象ステップのレコードを抽出する処理を実行する（ステップ2102）。このトレースデータから対象ステップのレコードを抽出する処理は、図19のステップ1914で指定された範囲に含まれるステップ番号を含まないトレースデータのレコードをトレースデータ用データエリア221から削除することにより行われる。

【0162】続いて、演算装置230は、呼出レコード番号を“0”に設定し（ステップ2103）、トレースデータ用データエリア221から当該呼出レコード番号のレコードを読み出し（ステップ2104）、次に、当該レコードのフラグB=1、すなわち、当該レコードのビットBが“1”かを調べる（ステップ2105）。

【0163】ここで、フラグB=1であると（ステップ2105YES）、読み出したレコードのステップの表示を活性状態にし（ステップ2106）、フラグB=0でないと（ステップ2105NO）、読み出したレコードのステップの表示を非活性状態にする（ステップ2107）。

【0164】ここで、ステップの表示を活性状態にする処理は、演算装置230により、表示制御用メモリ210を、トレースデータ用データエリア221のレコード中のフラグBが“1”であるステップ番号のステップに該当するデータを活性状態を意味するデータに書き換えることにより行われ、ステップの表示を非活性状態にす

る処理は、演算装置230により、表示制御用メモリ210を、トレースデータ用データエリア221のレコード中のフラグBが“0”であるステップ番号のステップに該当するデータを非活性状態を意味するデータに書き換えることにより行われる。

【0165】次に、当該レコードのフラグA=0、すなわち、ビットAが“0”かを調べ（ステップ2108）、ここで、A=0でないと（ステップ2108NO）、当該読出レコード番号を1インクリメントし（ステップ2110）、次に、当該読出レコード番号が総レコード数に達していないか、すなわち、「総レコード数>読出レコード番号」が成立するかを調べ（ステップ2111）、当該読出レコード番号が総レコード数に達していない場合は（ステップ2111YES）、ステップ2104に戻り、上記処理を繰り返す。

【0166】ステップ2108で、A=0であると判断され（ステップ2108YES）、ここで、オペレータにより入力装置270の、例えば、SPACEキーが押下されると、表示装置260の画面変化を停止し（ステップ2109）、当該読出レコード番号を1インクリメントし（ステップ2110）、次に、当該読出レコード番号が総レコード数に達していないか、すなわち、「総レコード数>読出レコード番号」が成立するかを調べ（ステップ2111）、当該読出レコード番号が総レコード数に達していない場合は（ステップ2111YES）、ステップ2104に戻り、上記処理を繰り返す。

【0167】そして、ステップ2111で、当該読出レコード番号が総レコード数に達し、「総レコード数>読出レコード番号」が成立しなくなると（ステップ2111NO）、このパターンBの解析実行処理が終了する。

【0168】なお、上記第1の実施の形態のデバッグ装置においては、PLCのCPUユニット100にサポートツール200を接続し、CPUユニット100とサポートツール200とが協働してトレース機能を用いたデバッグ処理を実現するように構成したが、上記処理をCPUユニット100のみにより行うように構成してもよい。

【0169】図22は、この発明に係わる工程歩進型プログラムのデバッグ装置の第2の実施の形態を示すブロック図である。

【0170】図22に示す第2の実施の形態の工程歩進型プログラムのデバッグ装置においては、PLCのCPUユニット100に、入力装置111および表示装置112を接続して構成される。

【0171】ここで、CPUユニット100は、入力装置111とのインタフェースをなすシリアルインタフェース（I/F）101、演算プロセッサ（CPU）102、入出力メモリ（I/Oメモリ）103、ユーザプログラムメモリ104、システムプログラムメモリ10

5、トレースメモリ106、ワークメモリ107、トレースパラメータメモリ108、表示装置用メモリ113、入出力装置(I/O)110とのインタフェースをなす入出力インタフェース(I/F)109、表示装置112とのインタフェースをなすインタフェース(I/F)114を具備して構成され、シリアルインタフェース101に入力装置111が接続され、入出力インタフェース109に入出力装置110が接続され、インタフェース(I/F)114に表示装置112が接続される。

【0172】ここで、CPU102は、このPLCユニット100の全体動作を統括制御するものである。

【0173】また、システムプログラムメモリ105には、CPU102で処理されるシステムプログラムが記憶され、ユーザプログラムメモリ104には、ユーザプログラムが記憶される。

【0174】また、トレースメモリ106には、このトレース機能を実現するためのサンプリング結果であるトレースデータが記憶される。

【0175】また、トレースパラメータメモリ108には、このトレース機能を実行するための各種トレースパラメータが記憶される。

【0176】このトレースパラメータメモリ108に記憶されたトレースパラメータとしては、

- 1) トレース対象チャート番号
  - 2) 接点およびステップ番号などのトレースのトリガ条件
  - 3) トレースのサンプリングディレイ値
- などが挙げられる。

【0177】ワークメモリ107には、CPU102の処理全般にわたる各種データが記憶される。

【0178】また、I/Oメモリ103は、CPU102によるプログラム実行により入出力装置110に対して入出力される入出力情報の代用として参照/書き換えられるメモリである。

【0179】また、表示装置用メモリ113は、表示装置112の表示を制御するためのものである。

【0180】また、入力装置111は、ユーザの要求を得るための装置で、キーボード、ポインティングデバイス等から構成される。

【0181】また、表示装置112は、表示装置用メモリ113の状態を表示する装置で、CRT、LCD等から構成される。

【0182】図23は、図22に示したCPUユニット100の全体処理フローを示すフローチャートである。

【0183】図22において、PLCの電源が投入(ON)されると、まず、所定の共通処理を実行し(ステップ2301)、次に、このPLCがプログラム運転可能かを調べる(ステップ2302)。ここで、プログラム運転可能と判断されると(ステップ2302YES)、

このPLCのプログラム運転処理を実行する(ステップ2303)。

【0184】また、ステップ2302でプログラム運転不可と判断された場合(ステップ2302NO)若しくはステップ2303のプログラム運転処理が終了すると、次に、I/Oメモリ103をリフレッシュするI/Oリフレッシュ処理を実行し(ステップ2304)、その後、周辺処理を実行し(ステップ2305)、ステップ2301に戻り、上記処理を繰り返す。

10 【0185】図24は、図22に示したCPUユニット100におけるトレースサンプリング処理フローを示すフローチャートである。

【0186】図24において、まず、該当ステップが活性化しているかを調べる(ステップ2401)。ここで、該当ステップが活性化していないと判断されると(ステップ2401NO)、該当ステップの非活性化処理を実行する(ステップ2402)。

【0187】そして、次に、該当ステップはトレース中かを調べ(ステップ2403)、トレース中であると判断されると(ステップ2403YES)、トレースポイントが示すトレースメモリ106のアドレスに、ビットBをOFF("0")にして該当ステップのステップ番号を格納し(ステップ2404)、トレースポイントを次に更新する(ステップ2408)。

【0188】また、ステップ2401において、該当ステップが活性化していると判断されると(ステップ2401YES)、該当ステップの活性化処理を実行する(ステップ2405)。

【0189】そして、次に、該当ステップはトレース中かを調べ(ステップ2406)、トレース中であると判断されると(ステップ2406YES)、トレースポイントが示すトレースメモリ106のアドレスに、ビットBをON("1")にして該当ステップのステップ番号を格納し(ステップ2407)、トレースポイントを次に更新する(ステップ2408)。

【0190】ステップ2408でトレースポイントを次に更新した後、若しくはステップ2403でトレース中ではないと判断された場合(ステップ2403NO)またステップ2406でトレース中ではないと判断された場合(ステップ2406NO)は、次に、残ステップはあるかを調べる(ステップ2409)。

【0191】ここで、残ステップがあると判断されると(ステップ2409YES)、ステップ2401に戻り上記処理が繰り返される。

【0192】ステップ2409で、残ステップがないと判断されると(ステップ2409NO)、次に、該当ステップはトレース中かを調べ(ステップ2410)、トレース中である場合は(ステップ2410YES)、トレースポイントが示すアドレスの1レコード前のレコードのビットAをONにして(ステップ2411)このト

レースサンプリング処理を終了する。

【0193】また、ステップ2410で、該当ステップはトレース中でないと判断されると(ステップ2410 NO)、このままこのレースサンプリング処理を終了する。

【0194】図25は、図23に示したCPUユニット100の周辺処理フローを示すフローチャートである。

【0195】図22に示したCPUユニット100は、

- 1) プログラム編集機能
- 2) この発明に係わるトレース機能
- 3) モニタ機能

等の多種の周辺機能をサポートしている。

【0196】そこで、図25に示したCPUユニット100の周辺処理フローにおいては、オペレータの選択操作により周辺各機能への分岐処理を実行する。

【0197】すなわち、図25において、まず、オペレータにより入力装置111から機能選択を行う(ステップ2501)。

【0198】そして、次に、ステップ2501でオペレータにより選択された機能はプログラム編集機能かを調べ(ステップ2502)、プログラム編集機能が選択されていると判断された場合は(ステップ2502 YES)、プログラム編集機能を実行する(ステップ2503)。

【0199】ステップ2502で、プログラム編集機能が選択されていないと判断された場合は(ステップ2502 NO)、次に、トレース機能が選択されたかを調べる(ステップ2505)。

【0200】ここで、トレース機能が選択されていると判断された場合は(ステップ2505 YES)、トレース機能を実行する(ステップ2506)。

【0201】また、ステップ2506でトレース機能が選択されていないと判断された場合は(ステップ2506 NO)、次に、モニタ機能が選択されているかを調べる(ステップ2507)。ここで、モニタ機能が選択されていると判断された場合は(ステップ2507 YES)、モニタ機能を実行し(ステップ2508)、モニタ機能が選択されていないと判断された場合は(ステップ2507 NO)、以下同様にしてこの周辺各機能への分岐を行い(ステップ2509)、この周辺処理フローを終了する。

【0202】なお、上記周辺処理フローは、周辺処理開始から終了までは、一定時間で中断し、次の周辺処理に置いてはその中断位置から再度継続される。

【0203】図26は、図22に示したCPUユニット100のトレース機能フローを示すフローチャートである。

【0204】ここで、この第2の実施の形態のデバッグ装置におけるトレースパラメータメモリ108の仕様およびトレースメモリ106の仕様について説明する。

【0205】この第2の実施の形態のデバッグ装置においては、図24に示したレースサンプリング処理と図25に示した周辺処理(トレース機能)とのインタフェース仕様として以下のフラグが用意されている。

【0206】1) トレース開始フラグ

このレース開始フラグは、周辺処理にてレース開始を依頼するときにONにされ、レースサンプリング処理側で認知されたときOFFにされる。

【0207】2) トレース中止(アボート)フラグは、周辺処理にてレースの中断を依頼するときONにされ、レースサンプリング側で認知されたときOFFにされる。

【0208】3) トレース完了フラグ

このレース完了フラグは、レースサンプリング側でサンプリング処理が完了したときONにされ、周辺処理にて新たにレースを開始するときOFFにされる。

【0209】なお、トレースメモリ106の仕様は、図16に示した第1の実施の形態のトレースメモリ106の仕様と同様である。

【0210】さて、図26に示すフローチャートの処理を、図22を参照して説明すると、まず、オペレータによる入力装置111からのトレースパラメータの入力が行われる(ステップ2601)。

【0211】ここでトレースパラメータとは、上述したように

- 1) トレースの対象
- 2) トレースの開始条件(トリガ、サンプリングディレイ値)

等である。このトレースパラメータは、CPUユニット100のトレースパラメータメモリ108に格納される(ステップ2602)。

【0212】次に、オペレータにより入力装置111からレース実行を選択し、このレース実行の指示は、CPUユニット100のトレースパラメータメモリ108に格納される(ステップ2604)。

【0213】次に、トレースパラメータメモリ108からレース完了フラグを読み出し(ステップ2605)、このレース完了フラグに基づきレース実行完了かを調べる(ステップ2606)。

【0214】ここで、レース実行完了と判断されると(ステップ2606 YES)、オペレータにより入力装置111からレース中断の入力を可能にする(ステップ2607)。

【0215】次に、オペレータによる入力装置111からの入力によりレース中断かを調べ(ステップ2608)、レース中断でないと(ステップ2608 NO)、ステップ2605に戻り、上記処理を繰り返すが、ステップ2608で、レース中断と判断されると(ステップ2608 YES)、レースアボートをトレースパラメータメモリ108に記憶し、このレース

機能フローを終了する。

【0216】また、ステップ2606で、トレース実行完了ではないと判断されると(ステップ2606N O)、SFCチャートの表示処理を実行する(ステップ2610)。このSFCチャートの表示処理は、トレースメモリ106のデータをユーザプログラムメモリ104のデータを元に表示制御用メモリ210にSFCチャートに相当するデータを書き込むことにより行われる。【0217】次に、オペレータによる入力装置111からのトレース表示範囲が選択される(ステップ2611)。そして、続いて、オペレータによる入力装置111からトレース結果表示のパターンがパターンAかパターンBかが選択される(ステップ2612)。【0218】次に、ステップ2612で、パターンAとパターンBとのどちらのパターンが選択されたかを調べ(ステップ2613)、ここで、パターンAが選択されていると判断されると、トレースメモリ106に格納されたトレースデータを用いてパターンAの解析表示処理を実行し(ステップ2614)、パターンBが選択されていると判断されると、トレースメモリ106に格納されたトレースデータを用いてパターンBの解析表示処理を実行し(ステップ2615)、このトレース機能フローを終了する。

【0219】なお、ステップ2614のパターンAの解析表示処理は、第1の実施の形態に関連して図20にフローチャートで示したパターンAの解析実行処理と同様であり、ステップ2615のパターンBの解析表示処理は、第1の実施の形態に関連して図21にフローチャートで示したパターンBの解析実行処理と同様である。

【0220】

【発明の効果】以上説明したようにこの発明によれば、プログラブルコントローラにおける工程歩進型プログラムの実行状態を監視するにおいて、上記プログラブルコントローラによる上記工程歩進型プログラムの実行時における各工程の活性化および/または非活性化の状態変化を示す情報を順次記憶し、上記記憶した上記工程歩進型プログラムの実行時における各工程の活性化および/または非活性化の状態変化を示す情報を読み出し、上記読出した上記工程歩進型プログラムの実行時における各工程の活性化および/または非活性化の状態変化を示す情報に基づいて、各工程の活性状態の変化を、図示化して表示された上記工程歩進型プログラム上にマッピングして表示するように構成したため、図示表示を利用して状態の変化を確認できる、工程歩進型プログラムの実行状態を直感的に認識できるという効果を奏する。

【0221】また、この発明によれば、上記読出した上記工程歩進型プログラムの実行時における各工程の活性化および/または非活性化の状態変化を示す情報に基づいて、各工程の活性状態の変化を、図示化して表示され

た上記工程歩進型プログラム上にマッピングして一括表示するように構成したので、上記の効果に加えて、工程歩進型プログラム中の分岐箇所をいずれに進行したかが一目で認識できるという効果を奏する。

【0222】また、この発明によれば、上記読出した上記工程歩進型プログラムの実行時における各工程の活性化および/または非活性化の状態変化を示す情報に基づいて、各工程の活性状態の変化を、図示化して表示された上記工程歩進型プログラム上にマッピングして変化順に表示するものであるように構成したので、上記の効果に加えて、活性状態の変化が発生した順をオペレータの望む時間感覚で再現でき、また、活性状態の遷移の狭間に、強制的な活性状態の変更が混入したとしてもそれを区別してかつ時系列的に確認できるという効果を奏する。

【図面の簡単な説明】

【図1】この発明に係わる工程歩進型プログラムのデバッグ装置の第1の実施の形態を示すブロック図である。

【図2】図1に示したサポートツール200の詳細構成を示すブロック図である。

【図3】この発明に係わる工程歩進型プログラムのデバッグ装置における工程歩進型プログラムの表示例を示す図である。

【図4】この発明に係わる工程歩進型プログラムのデバッグ装置におけるパターンAの表示例を示す図である。

【図5】この発明に係わる工程歩進型プログラムのデバッグ装置におけるパターンBの表示例を示す図である。

【図6】この発明に係わる工程歩進型プログラムのデバッグ装置におけるパターンBの表示例を示す図である。

【図7】この発明に係わる工程歩進型プログラムのデバッグ装置におけるパターンBの表示例を示す図である。

【図8】この発明に係わる工程歩進型プログラムのデバッグ装置におけるパターンBの表示例を示す図である。

【図9】この発明に係わる工程歩進型プログラムのデバッグ装置におけるパターンBの表示例を示す図である。

【図10】この発明に係わる工程歩進型プログラムのデバッグ装置におけるパターンBの表示例を示す図である。

【図11】この発明に係わる工程歩進型プログラムのデバッグ装置におけるパターンBの表示例を示す図である。

【図12】この発明に係わる工程歩進型プログラムのデバッグ装置におけるパターンBの表示例を示す図である。

【図13】この発明に係わる工程歩進型プログラムのデバッグ装置におけるパターンBの表示例を示す図である。

【図14】図1に示したデバッグ装置の全体動作を説明するための全体フローを示すフローチャートである。

【図15】図1に示したCPUユニット100の全体処

理フローを示すフローチャートである。

【図16】図1に示したCPUユニット100のトレースメモリ106の仕様を示す図である。

【図17】図1に示したCPUユニット100におけるトレースサンプリング処理フローを示すフローチャートである。

【図18】図1に示したサポートツール200の全体処理フローを示すフローチャートである。

【図19】図1に示したサポートツール200のトレース機能全体フローを示すフローチャートである。

【図20】図19示したパターンAの解析実行処理の詳細を示すフローチャートである。

【図21】図19示したパターンBの解析実行処理の詳細を示すフローチャートである。

【図22】この発明に係わる工程歩進型プログラムのデバッグ装置の第2の実施の形態を示すブロック図である。

【図23】図1に示したCPUユニット100の全体処理フローを示すフローチャートである。

【図24】図22に示したCPUユニット100におけるトレースサンプリング処理フローを示すフローチャートである。

【図25】図23に示したCPUユニット100の周辺処理フローを示すフローチャートである。

【図26】図22に示したCPUユニット100のトレース機能フローを示すフローチャートである。

【符号の説明】

\*

\* 100

CPUユニット

101

シリアルインタフェース(I/F)

102

演算プロセッサ(CPU)

103

入出力メモリ(I/Oメモリ)

104

ユーザプログラムメモリ

105

システムプログラムメモリ

106

トレースメモリ

107

ワークメモリ

108

トレースパラメータメモリ

109

入出力インタフェース(I/F)

110

入出力装置(I/O)

111

入力装置

112

表示装置

113

表示装置用メモリ

114

インタフェース(I/F)

200

サポートツール(デバッグ装置)

210

表示制御用メモリ

220

メモリ

221

トレースデータ用データエリア

222

ユーザプログラムデータエリア

223

サポートツールプログラムモジュール

230

演算装置(CPU)

240

バス

250

インタフェース(I/F)

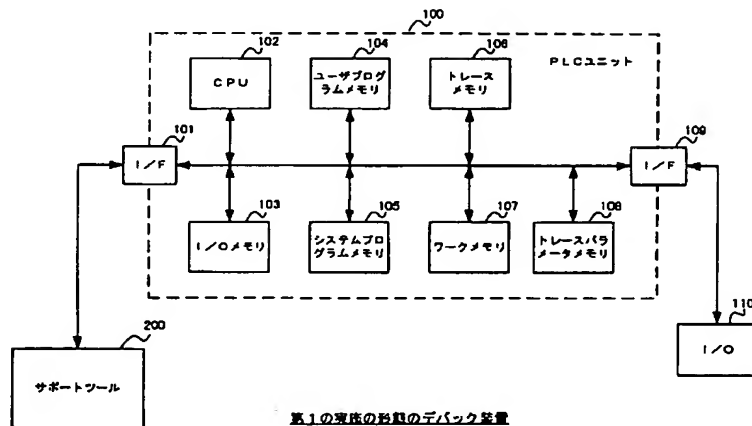
260

表示装置

270

入力装置

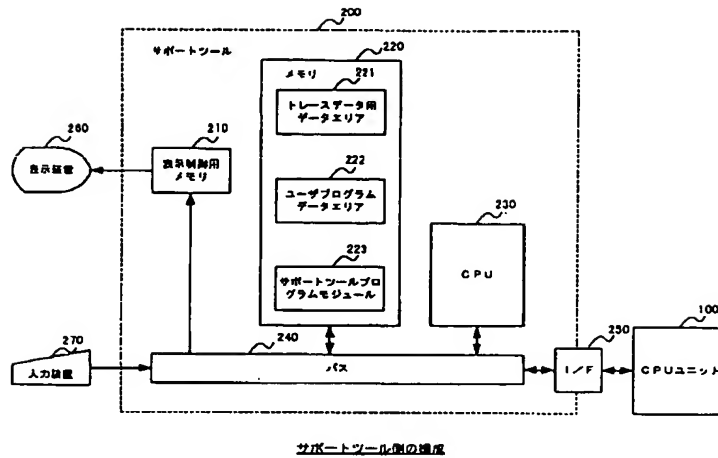
【図1】



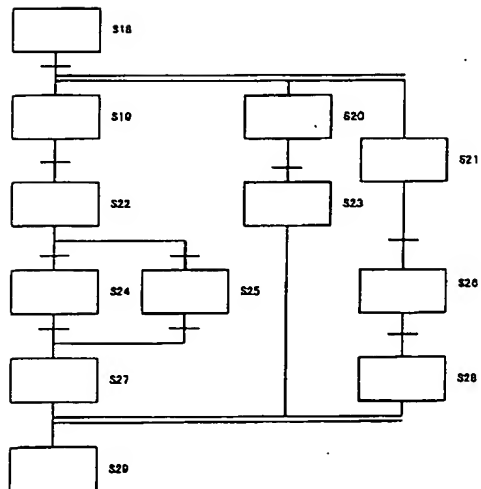
第1の実施の形態のデバッグ装置



【図2】

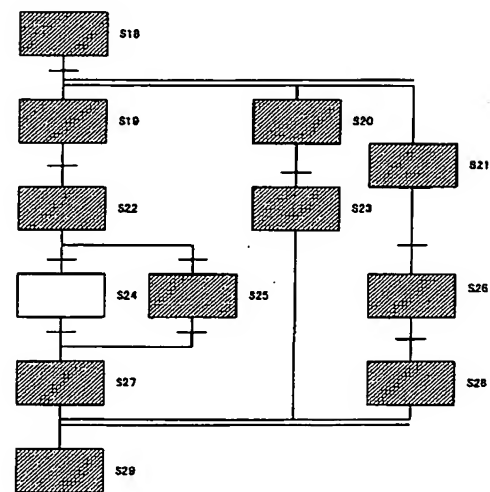


【図3】



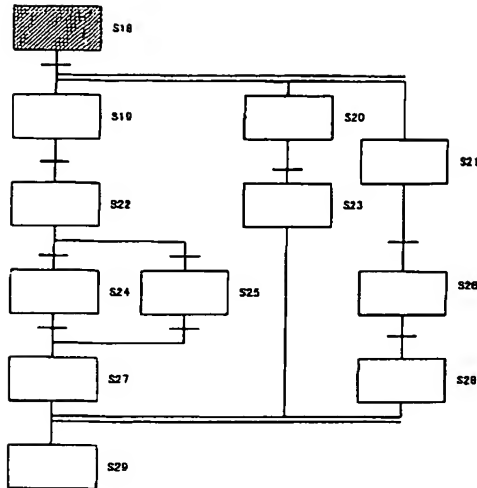
工程処理プログラムの表示例

【図4】



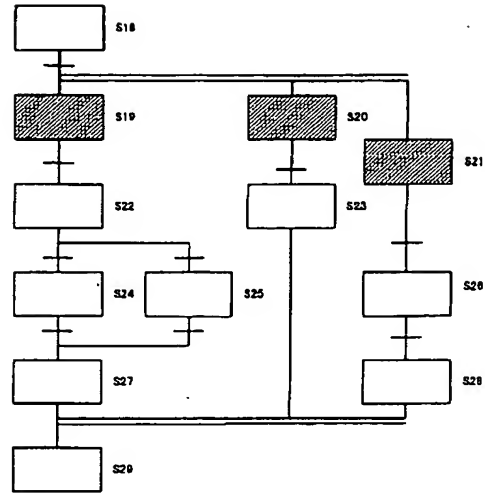
パターンAの表示例

【図5】



パターンBの表示例

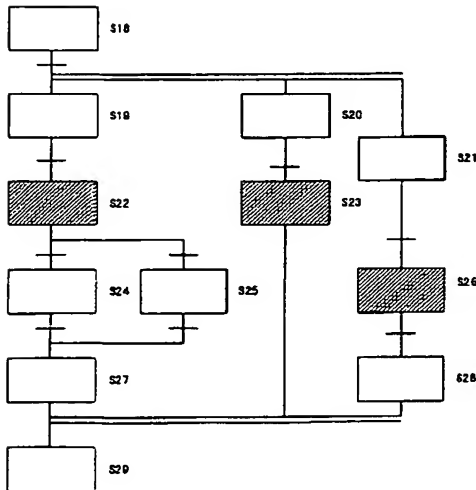
【図6】



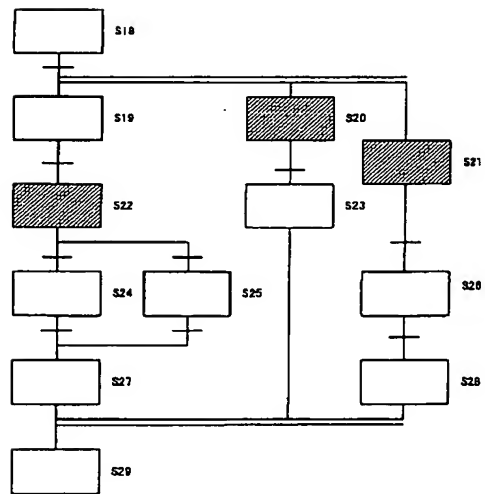
パターンBの表示例

【図7】

【図9】

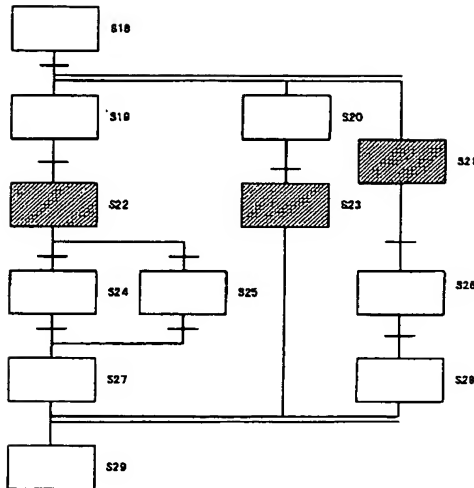


パターンBの表示例



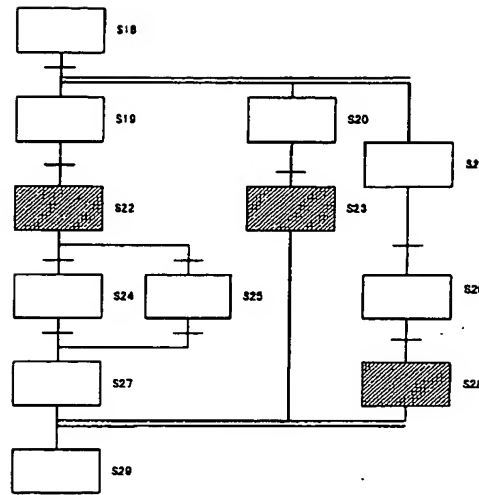
パターンBの表示例

【図8】



パターンBの表示例

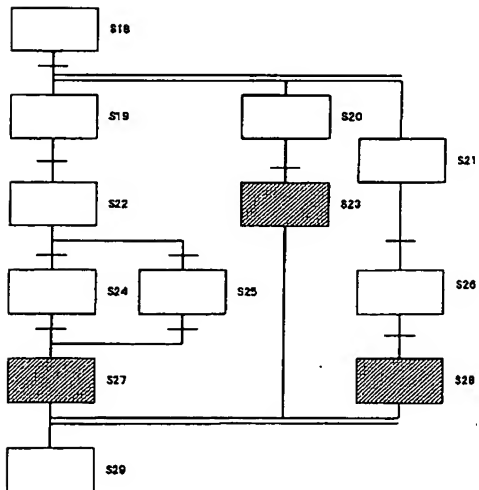
【図10】



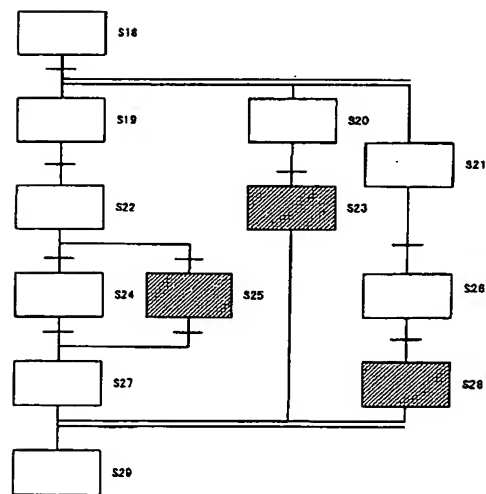
パターンBの表示例

【図11】

【図12】

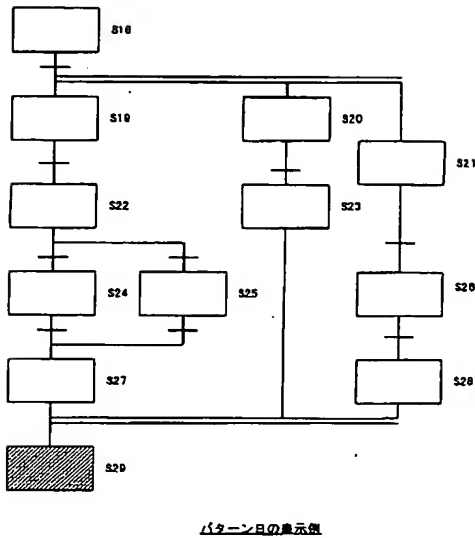


パターンBの表示例

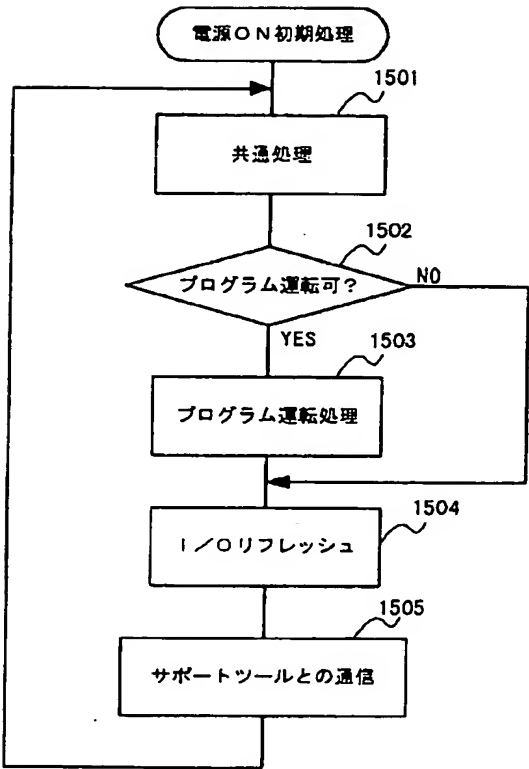


パターンBの表示例

【図13】



【図15】



【図16】

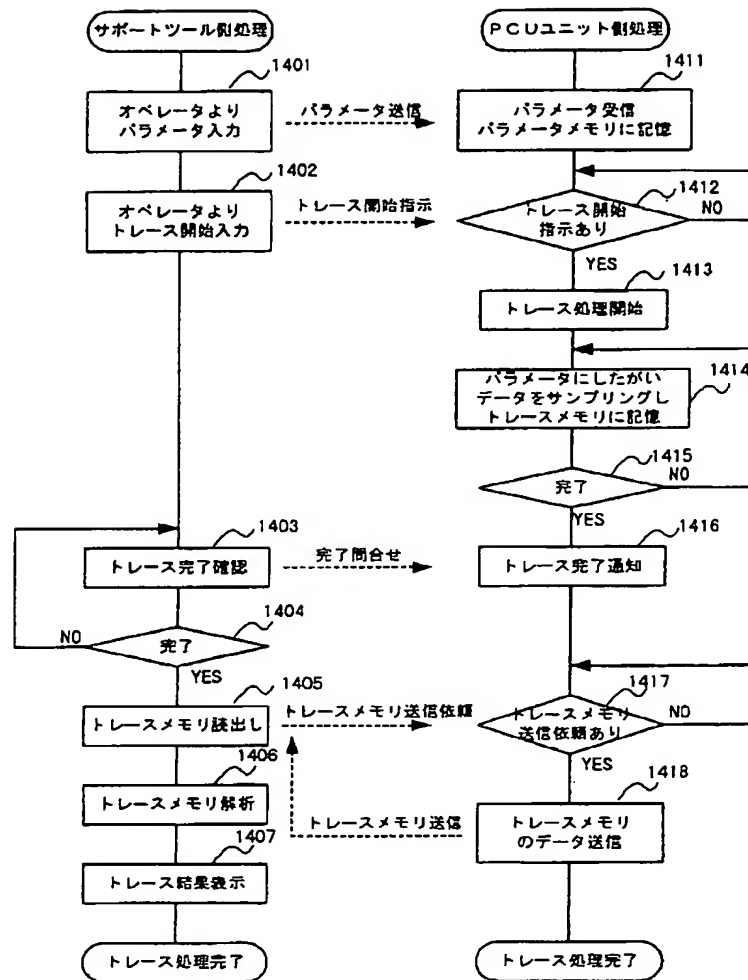
1 回のサンプリングデータ

A	B	
0	0	SFC STEP番号
1	0	SFC STEP番号
0	0	SFC STEP番号
0	0	SFC STEP番号
0	1	SFC STEP番号
0	1	SFC STEP番号
0	1	SFC STEP番号
1	1	SFC STEP番号
0	0	SFC STEP番号

トレースメモリ仕様

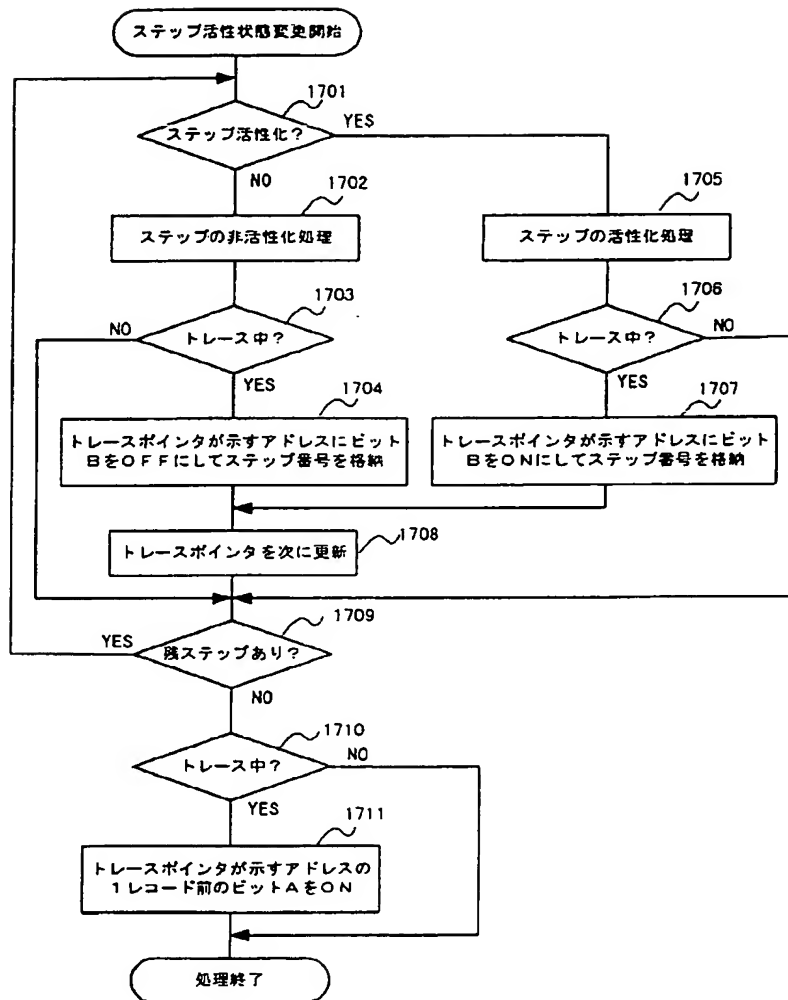
CPUユニット全体処理フロー

【図14】



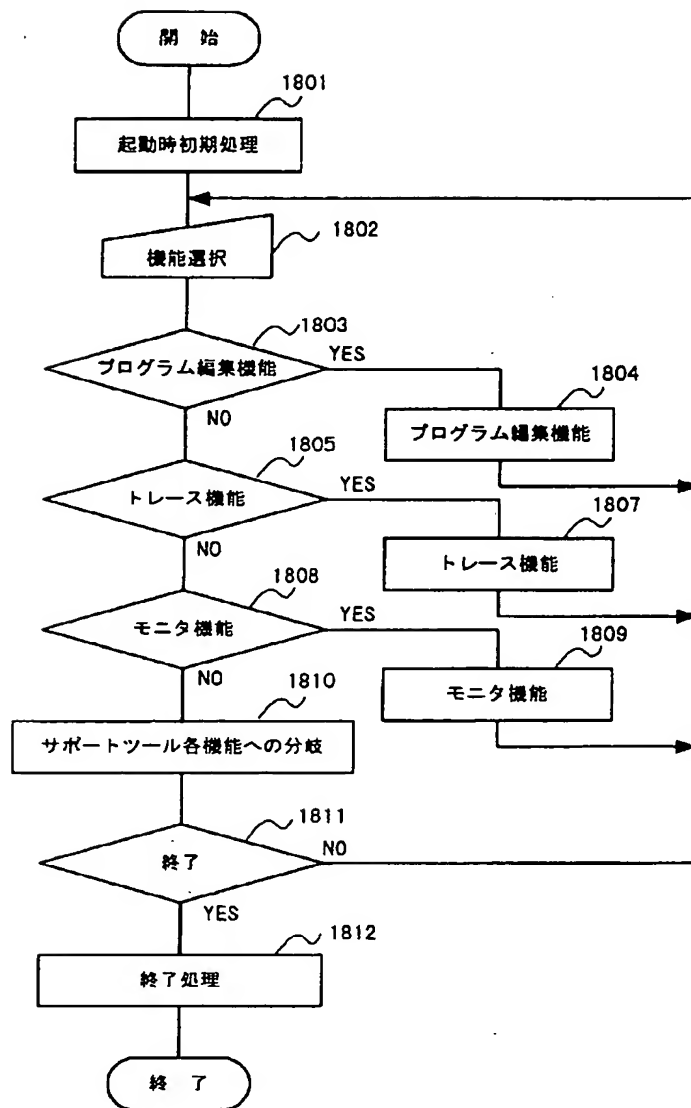
全体フロー

〔図17〕

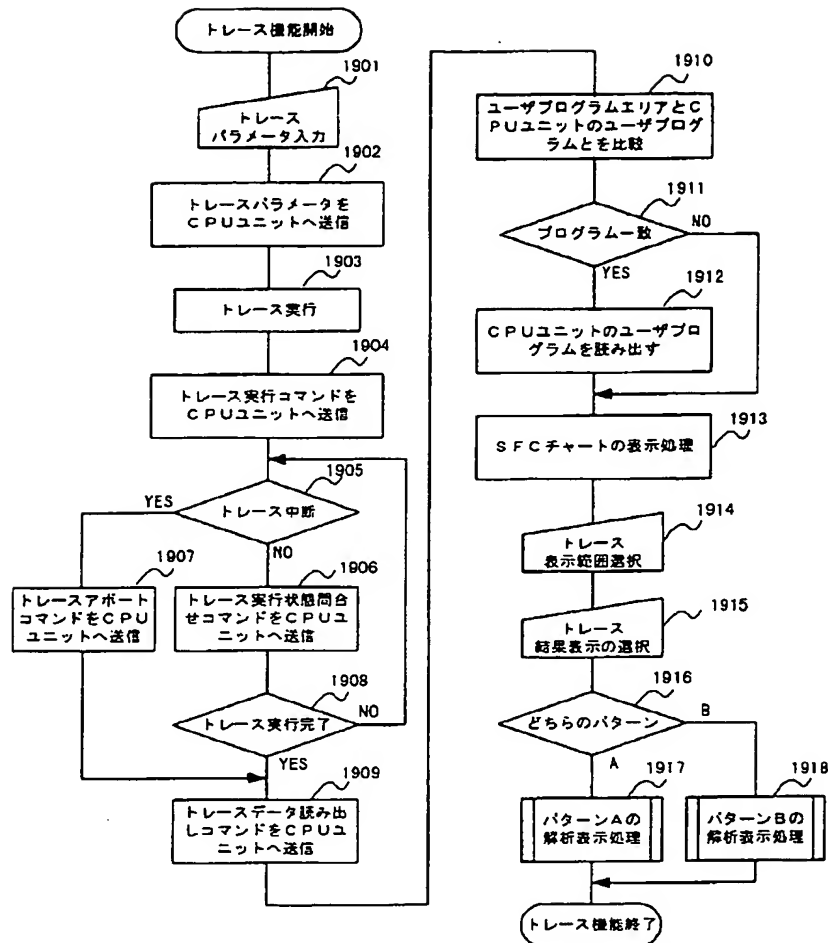


CPUユニット側トレースサンプリング処理フロー

【図18】

サポートツール全体フロー

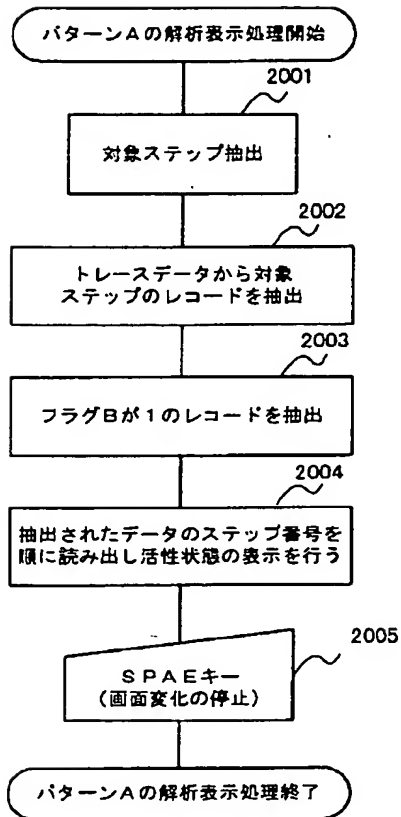
【図19】



トレース機能全体フロー

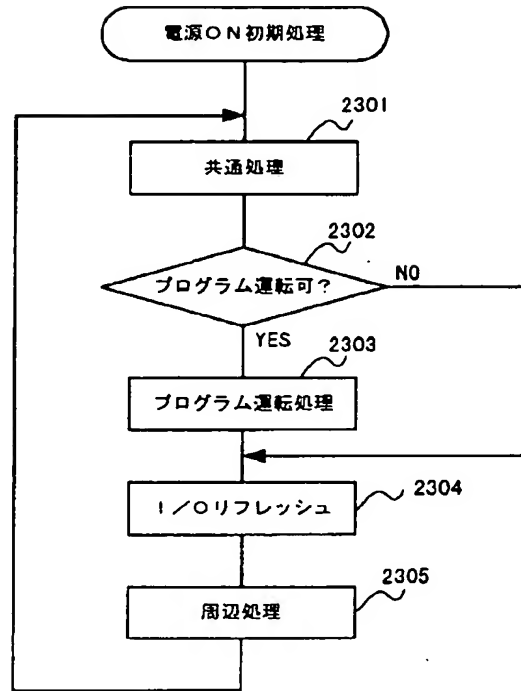


【図20】



トレースデータ解析表示フロー（パターンA）

【図23】



全体フロー

【図22】

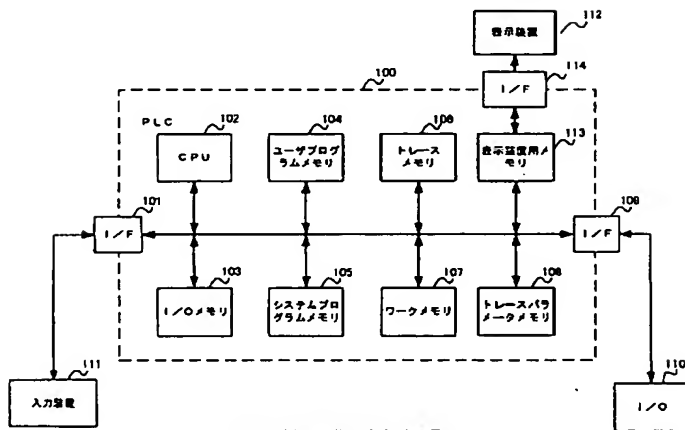
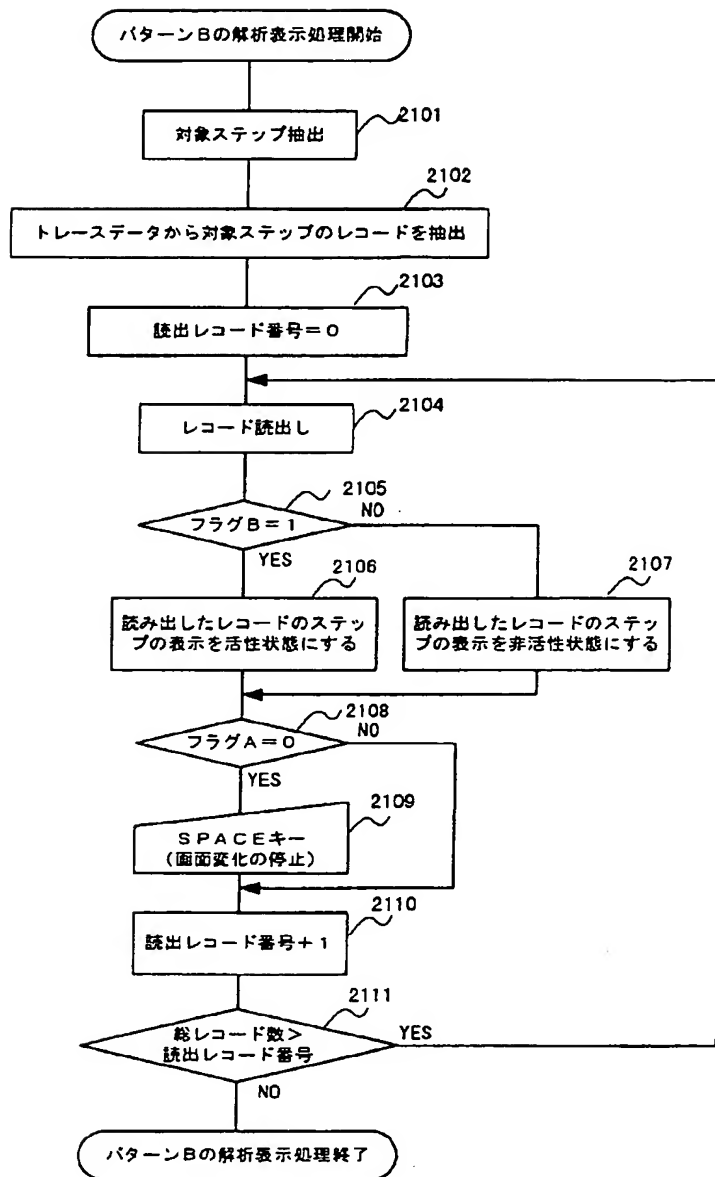


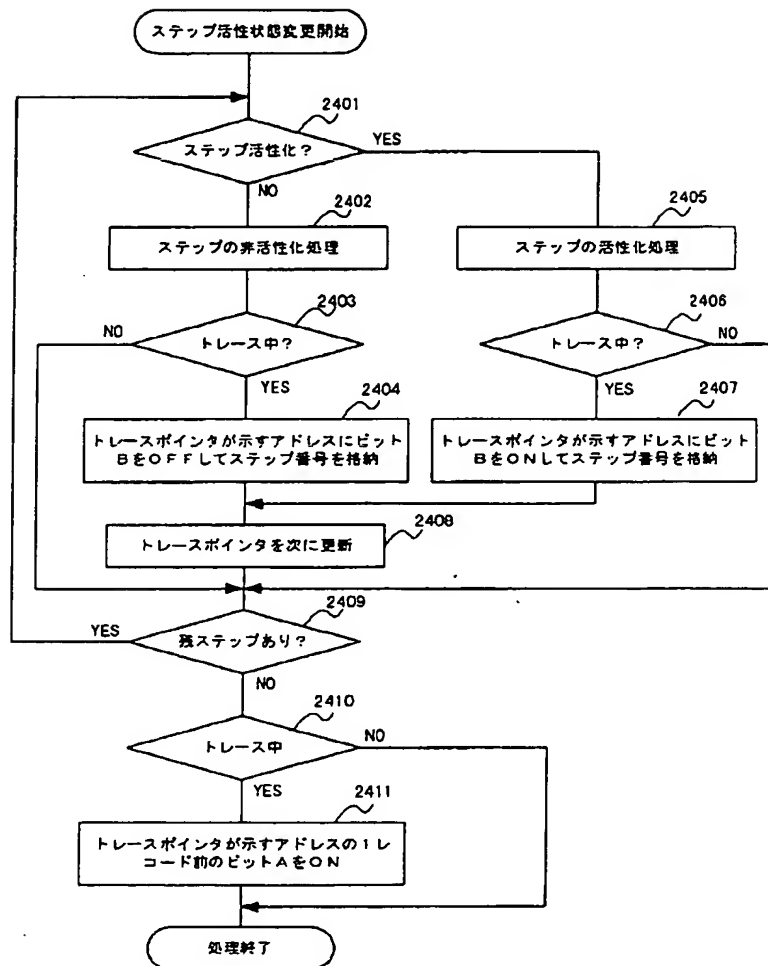
図2の要部の対象のバック装置

【図21】



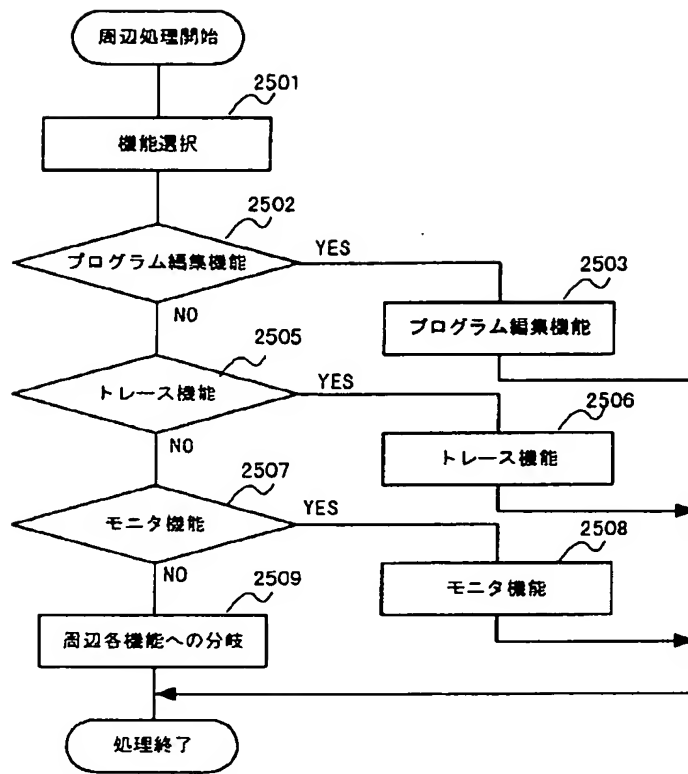
トレースデータ解析表示フロー（パターンB）

【図24】



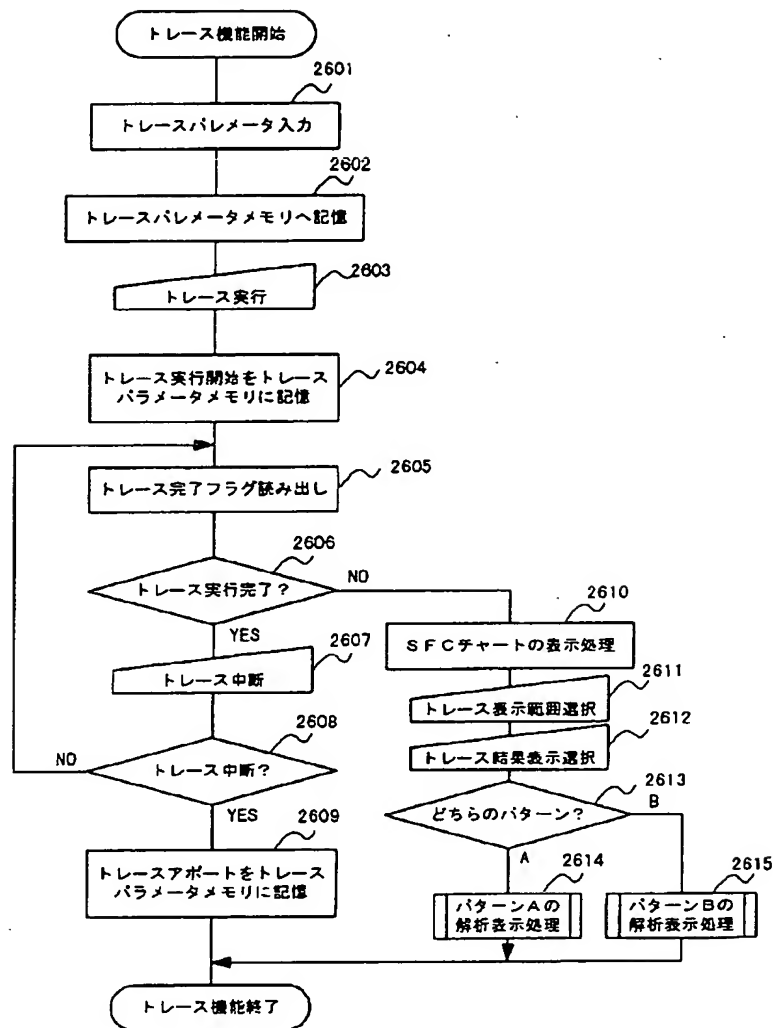
上レースサンプリング処理フロー

【図25】



周辺処理フロー

〔図26〕



トレース機能フロー

フロントページの続き

Fターム(参考) 5H220 BB10 CC05 CX06 DD07 JJ16  
 JJ19 JJ24 JJ27 JJ53 KK08  
 5H223 CC03 CC08 DD03 EE06 EE19  
 EE30 FF03